

基于大语言模型与Petri网的智能体服务组合方法研究

徐云哲¹, 杨凌霄¹, 张栋卓¹, 相东明^{2*}

¹浙江理工大学启新学院, 浙江 杭州

²浙江理工大学计算机科学与技术学院, 浙江 杭州

收稿日期: 2026年3月6日; 录用日期: 2026年4月16日; 发布日期: 2026年4月28日

摘要

随着大语言模型、智能体与服务计算技术的快速发展, 如何高效、可靠地组合多个智能体服务成为研究热点。Petri网因其严格的数学表达与强大的行为分析能力, 被广泛用于服务组合建模与验证。然而, 传统Petri网合成方法依赖人工定义语义匹配规则, 缺乏自动化与智能化支持。本文提出一种结合大语言模型与Petri网的智能体服务组合方法。首先, 提出面向智能体服务编排的UML建模方法, 统一描述服务流程; 其次, 设计UML模型向Petri网转换机制, 利用XMI与XSLT实现自动化映射; 进一步, 引入大语言模型对Petri网进行语义标注与智能融合, 实现基于语义的Petri网共享合成; 最后, 通过PIPE (Platform Independent Petri Net Editor)工具对合成后的Petri网进行模型检测, 验证其行为正确性。实验结果表明, 该方法能够有效提升智能体服务组合的自动化程度与可靠性。

关键词

智能体服务组合, Petri网, UML, 模型转换, 语义合成

Agent Service Composition Method Based on Large Language Models and Petri Nets

Yunzhe Xu¹, Lingxiao Yang¹, Dongzhuo Zhang¹, Dongming Xiang^{2*}

¹Qixin Honors School, Zhejiang Sci-Tech University, Hangzhou Zhejiang

²School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou Zhejiang

Received: March 6, 2026; accepted: April 16, 2026; published: April 28, 2026

Abstract

With the rapid development of Large Language Model (LLM), Agent and service computing

*通讯作者。

文章引用: 徐云哲, 杨凌霄, 张栋卓, 相东明. 基于大语言模型与 Petri 网的智能体服务组合方法研究[J]. 软件工程与应用, 2026, 15(2): 274-284. DOI: 10.12677/sea.2026.152026

technologies, how to efficiently and reliably compose multiple agent services has become a research hotspot. Petri nets are widely used for service composition modeling and verification due to their strict mathematical expression and strong behavioral analysis capabilities. However, traditional Petri net synthesis methods rely on manually defined semantic matching rules, lacking automation and intelligent support. This paper proposes an agent service composition method that combines large language models with Petri nets. First, a UML modeling method for agent service orchestration is proposed to uniformly describe service processes. Second, an XML-based mechanism for transforming UML models into Petri nets is designed, using XMI and XSLT to achieve automated mapping. Furthermore, large language models are introduced to semantically annotate and intelligently fuse Petri nets, enabling semantic-based shared synthesis of Petri nets. Finally, the synthesized Petri net is validated for behavioral correctness through model checking using the PIPE tool. Experimental results show that this method can effectively enhance the automation and reliability of agent service composition.

Keywords

Agent Service Composition, Petri Nets, UML, Model Transformation, Semantic Synthesis

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

智能体作为具有自主性、协作性与反应性的软件实体，在分布式系统、物联网、业务流程管理等场景中广泛应用。服务组合技术通过将多个子服务组合成复合服务，满足复杂业务需求。然而，智能体服务的异构性、动态性与交互复杂性给组合过程带来挑战，尤其是在行为一致性、语义匹配与自动化验证方面。

为应对上述挑战，研究者尝试引入形式化方法以增强服务组合的可靠性。Petri 网因其对并发、同步和资源共享等行为的精确刻画能力，成为建模智能体服务流程的重要工具。例如，Zhou 等人提出基于共享合成(Shared Composition)的 Petri 网构建方法，通过识别子网间的共享库所实现多智能体行为的协同建模，有效支持了并行程序与协同仿真系统的结构分析[1]。在虚拟维修训练等多人协同场景中，该方法被进一步用于合成多个操作者的 Petri 子网，实现了资源竞争控制与操作同步。

针对服务语义模糊与不确定性问题，Li 和 Wang 将模糊逻辑引入 Petri 网，构建了模糊 Petri 网模型，并系统地扩展了共享合成与同步合成运算；他们利用关联矩阵建立了合成网与原子网之间的代数关系，为处理含不确定信息的智能体服务组合提供了理论支撑[2]。此外，A. Mahadeer 等人提出部分 Petri 网(Partial Petri Net)概念，通过定义其关联语言及闭包特性，为不确定分布式系统的验证、确认与合成提供了精准技术，拓展了 Petri 网在含“部分可观测”特性的智能体服务组合中的应用场景[3]。

在动态组合机制方面，Chen 等人设计了一种基于策略驱动的智能体服务组合框架，使系统能够根据环境变化自主选择与重组服务，显著提升了组合过程的适应性与智能化水平[4]。Jorg Desel 与 Wolfgang Reisig 则围绕基本 Petri 网的合成问题展开研究，提出通过“区域”概念从转换系统构造 Petri 网的方法，证明了存在多项式规模的 Petri 网与给定转换系统对应，为服务组合的模块化构建提供了理论基础[5]。Ronny Tredup 针对 b-有界 Petri 网的合成复杂性进行了全面分析，为不同场景下的服务组合方法选择提供了复杂性参考[6]。

然而, 尽管上述方法在结构建模上取得进展, Zhang 等人指出, 现有 Petri 网合成技术仍严重依赖人工预定义的接口匹配规则, 缺乏对服务功能语义的深度理解, 难以实现端到端的自动化验证与合成[7]。

综上所述, 当前智能体服务组合虽已借助 Petri 网在形式化建模方面取得一定成果, 但在语义驱动的自动合成、跨异构服务的行为一致性保障以及与自然语言需求的无缝对接等方面仍存在明显瓶颈。近年来, 大语言模型(LLM)展现出强大的语义理解与逻辑生成能力, 为突破这些限制提供了新可能。本文旨在融合 LLM 与 Petri 网合成机制, 提出一种面向智能体服务的语义感知、自动可验证的组合方法。

针对上述问题, 本文提出一种基于大语言模型与 Petri 网的智能体服务组合方法。具体包括:

- (1) 提出面向智能体服务编排的 UML 建模方法, 统一描述服务行为与交互;
- (2) 设计基于 XML 的 UML 模型向 Petri 网转换机制, 实现自动化映射;
- (3) 引入大语言模型对 Petri 网进行语义标注与智能融合;
- (4) 基于 PIPE 工具对合成后的 Petri 网进行模型检测与验证。

2. 相关知识

2.1. Petri 网

Petri 网是对离散并行系统的数学表示[8], 适合于描述异步的、并发的计算机系统模型。Petri 网既有严格的数学表述方式, 也有直观的图形表达方式, 既有丰富的系统描述手段和系统行为分析技术, 又为计算机科学提供坚实的概念基础。

一个经典的 Petri 网[9] [10]由四元组构成, 通常表示为 $PN = (P, T, F, M_0)$, 其中:

- P 是库所(Place)的有限集合;
- T 是变迁(Transition)的有限集合, 且 $P \cap T = \emptyset$;
- F 是有向弧的集合, 满足 $F \subseteq (P \times T) \cup (T \times P)$, 用于描述库所与变迁之间的连接关系;
- M_0 是初始标识(Marking), 表示系统初始状态下各库所中令牌(Token)的分布情况。

2.2. Petri 网合成

Petri 网合成是指将多个子 Petri 网 $N_i = (P_i, T_i, F_i, M_{0i})$ (其中 $i \in \{1, 2, \dots, k\}$), 通过合并共享变迁(比如语义相似的变迁), 构建一个复合 Petri 网的过程。设共享变迁集 $T' \subseteq \bigcup_{i=1}^k T_i$, 合并得到的 Petri 网为 $N_G = (P_G, T_G, F_G, M_{0G})$, 其构造规则如下:

- $T_G = \bigcup_{i=1}^k T_i$, 共享变迁仅保留一份;
- $P_G = \bigcup_{i=1}^k P_i$ 且满足 $P_i \cap P_j = \emptyset$ ($i \neq j$);
- $M_{0G}(p) = M_{0i}(p)$;
- 流关系 F_G 由所有子网中的弧经变迁重定向后合并得到, 即若在子网 N_i 中存在 t 与 p 之间的有向弧, 则在合成网中保留该弧(若 t 为共享变迁, 则对应合并后的变迁)。

3. 基于大语言模型与 Petri 网的智能体服务组合方法

本文提出的智能体服务组合方法遵循如图 1 所示的技术路线图, 涵盖四个核心阶段。首先, 在输入层使用 UML 活动图和时序图对智能体服务流程进行建模, 并导出为标准 XMI 文件; 其次, 在转换层基于 XSLT 转换方法将 UML 模型自动转换为 Petri 网; 接着, 通过大语言模型合成模块, 利用大语言模型对 Petri 网进行语义标注与智能融合, 实现共享合成; 最终, 在输出层利用 PIPE 工具对合成的 Petri 网进行形式化验证, 确保智能体服务组合的正确性与可靠性, 完成从设计、转换、合成到验证的完整技术链条。

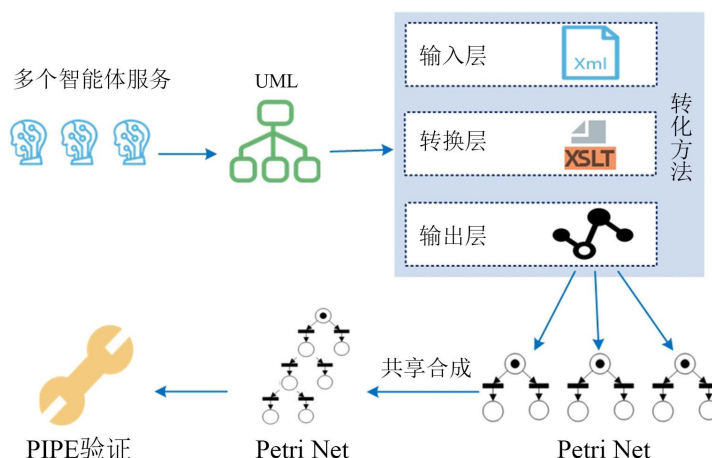


Figure 1. Technology roadmap
图 1. 技术路线图

3.1. 面向智能体服务编排的 UML 建模方法

3.1.1. 手动生成 UML 活动图的建模方法

手动生成 UML 活动图是一种人工主导、逻辑驱动的建模方式，核心在于精准拆解流程环节、梳理步骤间的关联关系。

建模时，首先要明确活动图的描述对象与边界，区分需要重点呈现的核心流程和无需纳入的外部关联环节，同时确定流程的责任主体以规划泳道划分。随后，拆解流程中的关键 UML 元素，包括代表核心任务的活动、不可拆分的原子化动作、表示步骤顺序的控制流，以及用于条件判断的分支节点、用于并行操作的分叉节点等。完成元素拆解后，按时间顺序或逻辑顺序排列各步骤，标注分支判断的触发条件，识别可并行执行的环节，最后借助工具绘制图形。

3.1.2. 自动 UML 生成活动图的建模方法(大语言模型为主)

自动生成 UML 活动图是依托算法、工具或数据解析实现的高效建模方式，通过输入结构化信息让工具自动提取流程逻辑并生成图形。

以大语言模型为核心的自动生成活动图方法，是通过自然语言理解与逻辑推理能力，将流程文本直接转化为规范 UML 活动图的高效路径。使用时，只需输入清晰的流程描述文本，并添加简单的 prompt 约束，比如明确参与者、分支条件、并行环节等关键信息，模型就能自动解析文本语义，识别并映射 UML 活动图的核心元素。大语言模型输出的结果通常有两种形式，一种是可直接渲染的 PlantUML 代码，复制到对应工具就能生成可视化图形；另一种是结构化的元素清单，导入 Draw.io 等平台即可快速组装。

3.2. 智能体服务的 UML 模型到 Petri 网模型转换方法

3.2.1. 转换框架设计

本文设计了基于 XML 技术栈的三层转换框架，实现了从智能体服务的活动图到 Petri 网的标准化、自动化转换流程。

(1) 输入层负责与 UML 建模工具对接。通过 XMI (XML Metadata Interchange) 标准实现模型数据的统一采集，支持主流 UML 工具导出的 XMI 2.x 版本文件。无论使用 Enterprise Architect、StarUML 还是 ArgoUML，只要支持标准 XMI 导出，均可作为本框架的输入源，确保了良好的工具兼容性。

(2) 转换层是框架的核心模块, 采用 XSLT 技术实现模型语义的精确映射。该层包含三个关键组件: 模型类型检测器自动识别输入 XMI 文件对应的活动图; XSLT 样式表库包含针对活动图的专用转换规则。

(3) 输出层生成符合 ISO/IEC 15909 标准的 PNML 文件。PNML 是基于 XML 的 Petri 网描述语言, 具有丰富的工具生态支持。生成的 PNML 文件可直接导入 PIPE、CPN Tools、Ptolemy 等主流 Petri 网分析工具进行可视化、仿真与形式化验证。

该框架的核心优势在于标准化和松耦合: 各层之间通过标准 XML 格式衔接, 易于替换或升级单个组件, 同时保证了转换过程的自动化与可靠性。

3.2.2. 映射方法设计

UML 活动图的核心语义是活动间的控制流传递。本算法采用“控制流驱动”的映射策略, 通过如表 1 所示的规则将 UML 活动图元素映射为 Petri 网元素。

Table 1. Mapping rules from UML activity diagram elements to Petri net elements
表 1. UML 活动图元素到 Petri 网元素的映射规则

UML 活动图元素	Petri 网元素	映射说明
初始节点	库所(含令牌)	表示流程开始状态
活动节点	变迁	表示可执行的动作
决策节点	库所 + 多输入变迁	输出选择条件
合并节点	多输出变迁 + 库所	汇聚多个控制流
分叉节点	库所 + 变迁分裂	产生并发分支
汇合节点	变迁合并 + 库所	同步多个并发分支
终止节点	库所(吸收令牌)	表示流程结束
控制流边	弧	连接库所与变迁

3.3. 基于大语言模型的 Petri 网动态共享合成

定义 1: 语义关联度函数 设 $D(x)$ 为对象 x 关联的自然语言描述文档, S_{LLM} (Semantic Large Language Model) 为一个能够计算两个文本语义相似度的大语言模型。

$$Sim(x, y) = S_{LLM}(D(x), D(y)) \in [0, 1];$$

该函数用于计算两个元素 x, y (如库所、变迁或整个子网) 之间的语义相似度。

定义 2: 带语义标记的 Petri 网 (Semantic-Labeled Petri Net, SLPN) 智能体服务对应的模型被定义为一个带语义标记的 Petri 网 $N = (P, T, F, M_0, O, Doc)$; 其中,

- (P, T, F, M_0) 为 Petri 网;
- **语义标记函数** $O: T \rightarrow Doc$ 将每个变迁映射到其关联的自然语言描述文档(或语义描述文档); 例如, 变迁可能描述“处理订单”;
- Doc 为整个子网 N 对应的全局自然语言描述文档(用于宏观语义理解)。

3.3.1. 语义关联与可融合判定

为了实现两个智能体服务(子网 N_a 和 N_b) 的协同工作, 需要识别它们之间的连接点(即变迁对)。为此, 现给出变迁的可融合性条件。

定义 3: 变迁对的可融合性(Fusability of Place Pairs)给定两个变迁 $ta \in Ta$ 和 $tb \in Tb$ ，它们被视为可融合的(Fusable)，当且仅当满足：

$$Fusable(t_a, t_b) \Leftrightarrow Sim(t_a, t_b) \geq \theta_P$$

其中， $Sim(ta, tb) \geq \theta_P$ 为**语义关联条件**。 θ_P 是预定义的变迁语义相似度阈值，表示两个变迁描述的语义足够接近。例如， ta 描述“用户支付信息”， tb 描述“客户付款详情”。

在满足可融合性条件 $Fusable(ta, tb) = TRUE$ 的前提下，可对 ta 和 tb 执行融合操作生成新的融合子网 N_{fused} ，具体包括节点融合与结构合并：节点融合将 ta 和 tb 合并为一个新变迁 $tfused$ ，其语义 $O(tfused)$ 通过调用 $FuseSemantics(ta, tb)$ 利用 LLM 对 $O(ta)$ 和 $O(tb)$ 进行融合生成(例如生成“合并支付信息节点”)；结构合并则保留 Na 和 Nb 中除 ta 、 tb 外的所有节点和流关系，并将原指向 ta 或 tb 的弧全部重定向至 $tfused$ ，重复以上操作直至所有满足可融合性条件的库所对均完成融合，最终得到两个完全融合的含语义 Petri 网。

3.3.2. 动态共享合成算法

基于语义关联与可融合判定条件，本文给出智能体服务动态共享合成的基本算法，如算法 1 所示。

该共享合成算法以两个带语义标记的 Petri 网为输入，通过语义相似度阈值筛选可融合的变迁对进行迭代融合，最终输出一个统一的融合网；算法首先通过合并操作将两个网的所有库所、变迁、弧及初始标识整合为初始融合网，并利用 $findSimilarPairs$ 函数遍历所有跨网变迁对，基于 sim 函数计算其语义描述的相似度，筛选出满足阈值条件的候选对集合；随后进入主循环，每次通过 $getPair$ 函数取出一个候选对，首先调用 $isFusable$ 函数检查两个变迁在结构上是否兼容，若通过则执行融合操作：由 $createTransition$ 创建新变迁节点， $mergeSemantic$ 函数调用大语言模型将原变迁的语义描述融合生成新变迁的语义标记， $redirectArcs$ 函数将所有指向原变迁的输入输出弧重定向至新变迁，之后从网中删除原变迁并将新变迁加入变迁集，最后调用 $updateCandidates$ 函数从候选集中移除涉及已融合变迁的所有对，并基于新变迁与剩余变迁的语义相似度，将满足阈值的新对补充进候选集以支持链式融合；算法重复此过程直至候选集为空，此时返回的融合网即为最终结果。

算法 1: 智能体服务动态共享合成算法

输入: 语义 Petri 网 $N_A = (P_A, T_A, F_A, M0_A, O_A, Doc_A)$

$N_B = (P_B, T_B, F_B, M0_B, O_B, Doc_B)$

相似度阈值 θ

输出: 融合网 N

$N = N_A \cup N_B$ // 初始化，合并两个网的所有元素。

// 找出所有语义相似的变迁对

$C = findSimilarPairs(T_A, T_B, \theta)$

While $C \neq \emptyset$: // 循环融合所有可融合的相似对

$(ta, tb) = getPair(C)$

If $isFusable(ta, tb)$:

$tf = createTransition()$ // 创建新变迁

// 融合语义描述

$O[tf] = mergeSemantic(O[ta], O[tb])$

$redirectArcs(ta, tb, tf)$ // 重定向所有弧到新变迁

$removeTransitions(ta, tb)$ // 删除旧变迁

$N.T.add(tf)$
 $C = updateCandidates(C, tf, ta, tb) //$ 更新候选集

Return N

例如，在如图 2 中的 Petri 网 N1、N2 中，变迁 T3 与 T6 存在语义关联且相似度 $Sim(T_3, T_6) \geq \theta_p$ ，符合变迁合并原则，从而生成 Petri 网 N'。

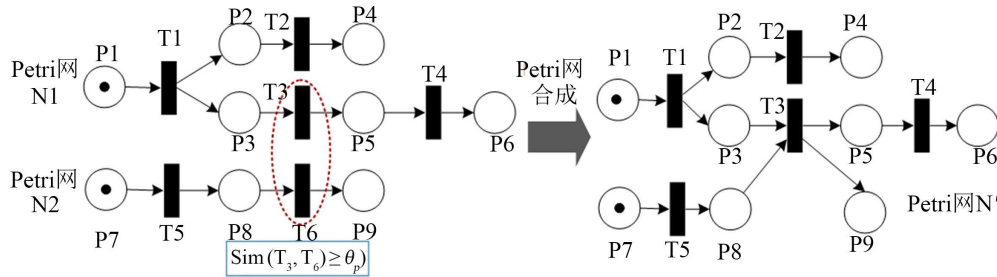


Figure 2. Petri net shared synthesis
图 2. Petri 网共享合成

3.4. 基于 PIPE 的智能体服务模型检测

(1) 有界性与安全性分析

有界性与安全性分析是 PIPE 工具用于验证 Petri 网模型资源可控性的功能，分为有界性分析和安全性分析两个子项。其中有界性分析通过检测每个库所的 Token 数量是否存在上限，若某库所被判定为“无界”，则说明对应的 UML 活动图存在无限循环的逻辑漏洞，比如“密码验证失败”后未设置重试次数上限，导致流程反复触发该环节，Token 持续堆积；安全性分析则通过验证所有库所的 Token 数量是否始终不超过 1，若不满足安全性，意味着 UML 流程中存在并发冲突风险，比如多个并行流程同时向同一个状态节点发送信号，需调整 UML 活动图的并行逻辑以规避冲突。

(2) 死锁检测

死锁检测是 PIPE 工具保障 Petri 网模型流程稳定性的关键功能，专门识别模型中“Token 停滞、无任何变迁可触发”的阻塞状态，对应 UML 活动图中流程卡死的隐患，如表 2 所示。执行检测时通过扫描所有可达状态，筛选出满足死锁条件的状态，并列出现该状态下的库所标记情况。这些死锁状态往往对 UML 活动图的逻辑缺陷，比如并行分叉后的某条分支缺少对应的结束节点，导致汇合变迁因输入库所 Token 不全无法触发，或是分支判断的条件覆盖不完整，出现无后续流程的“孤儿”状态。通过死锁检测结果，可精准回溯到 UML 活动图的对应环节进行修正。

Table 2. Comparison table of PIPE test results and UML activity diagram corrections
表 2. PIPE 检测结果与 UML 活动图修正对应对照表

PIPE 检测类型	典型异常结果	对应的 UML 活动图逻辑缺陷
有界性分析	某库所被判定为“无界”	UML 活动图存在无终止条件的循环
安全性分析	某库所 Token 数量超过 1	1. 并行流程同时向同一个状态节点发送信号，引发并发冲突 2. 变迁触发规则设置错误，导致重复生成 Token
死锁检测	出现 Token 停滞的死锁状态	1. 并行分叉后部分分支无结束节点，导致汇合节点无法触发 2. 分支判断的条件覆盖不全，存在“无匹配条件”的空白分支 3. 活动节点缺少后续流程，成为孤立步骤

4. 案例分析

场景描述: 用户通过平台下单后, 系统需协调“商家智能体”处理订单、“骑手智能体”执行配送, 并实时同步状态给“用户智能体”。各智能体服务由不同供应商提供, 描述异构, 需动态组合。

4.1. 智能体服务 UML 建模

对以上智能体进行 UML 建模, 如图 3 所示。

(1) 骑手智能体活动图(图 3(a)): 骑手上线后监听订单, 确认接单后前往商家, 核实餐品就绪状态后取餐, 配送至用户地址并完成交付, 同步通知至相关智能体, 流程终止。

(2) 商家智能体活动图(图 3(b)): 接收平台转发的订单后, 核对详情并确认是否接单, 接单后启动餐品制作, 完成后标记出餐状态并通知骑手, 待骑手取餐确认后流程结束; 若拒绝接单则同步原因至用户, 流程终止。

(3) 用户智能体活动图(图 3(c)): 用户启动平台后浏览菜单, 可选择继续浏览或加入购物车, 选定菜品后完成地址选择与支付操作。支付成功则进入订单等待状态, 实时接收商家与骑手的状态同步; 支付失败可选择重新支付或取消订单, 流程终止。

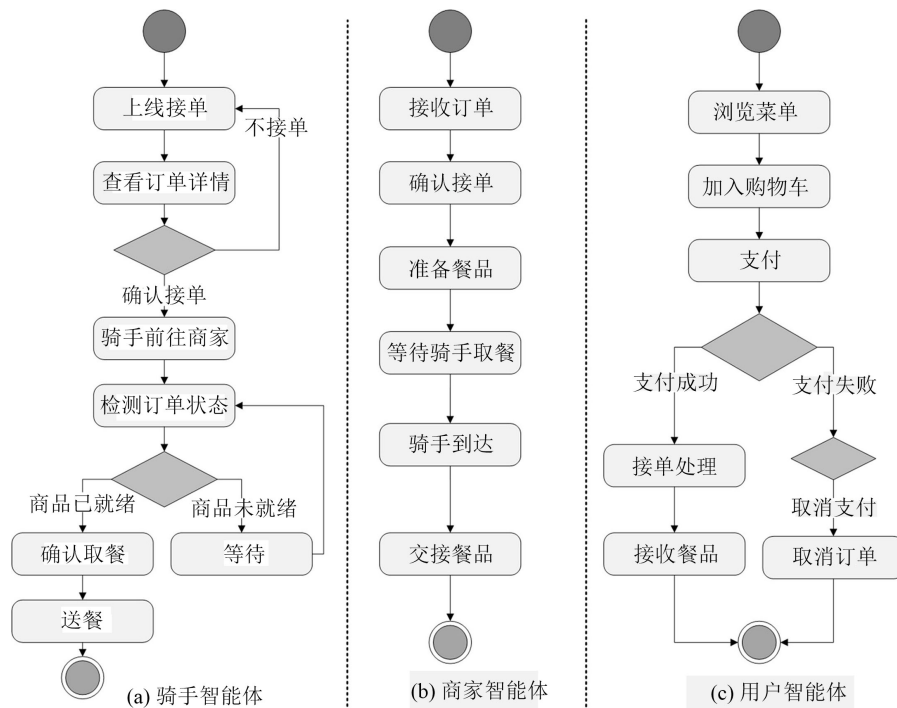


Figure 3. Agent activity diagram

图 3. 智能体活动图

4.2. 从 UML 模型到 Petri 子网的自动转换

针对“外卖订单协同处理”场景下的三个智能体服务(用户、商家、骑手) UML 活动图为输入, 通过自动化转换流程生成对应的 Petri 网子网模型。转换过程采用基于 XMI 与 XSLT 的技术框架, 将 UML 活动图元素按既定规则映射为 Petri 网元素, 确保语义一致性。依据转换规则, 成功生成 3 个 Petri 子网, 如图 4 所示, 其分别对应“商家智能体”、“骑手智能体”和“用户智能体”的服务流程。将生成的 PNML 文件导入 PIPE 工具进行初步仿真验证, 确认各子网结构完整、无孤立节点, 且从初始状态到终止状态存在可达路径。验证

结果表明，转换过程保持了 UML 模型的语义与逻辑完整性，为后续语义融合与合成提供了结构化基础。

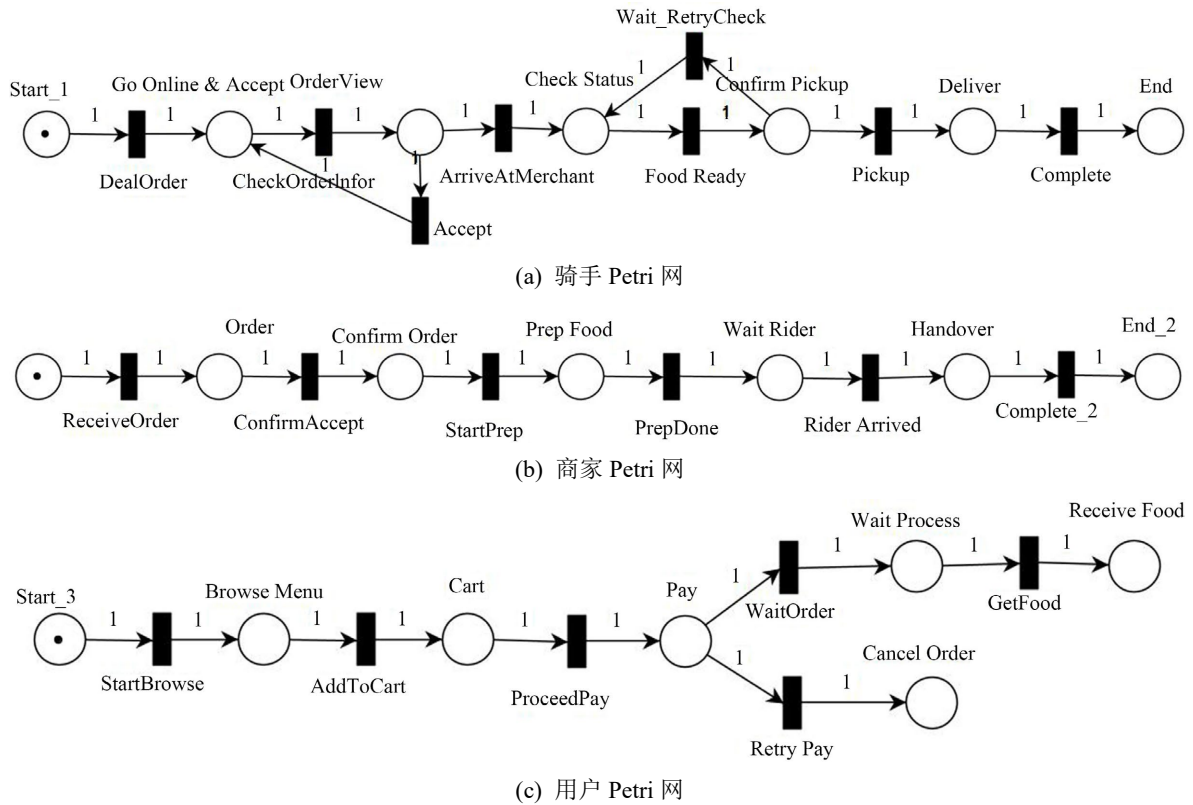


Figure 4. Agent service Petri net
图 4. 智能体服务 Petri 网

4.3. 基于大语言模型的 Petri 网合成

本论文通过 LLM (DeepSeek)对骑手、商家、用户三个智能体子网进行语义解析与相似度计算，识别跨子网的共享变迁，为后续 Petri 网融合奠定基础。表 3 展示了四组跨子网变迁对的语义相似度计算结果 (阈值 $\theta \geq 60\%$)。

Table 3. Petri net synthesis connection table
表 3. Petri 网语义相似度

序号	变迁 1	语义 1	变迁 2 素	语义 2 型	相似度(LLM-DeepSeek)
1	T_a1 (DealOrder)	上线接单	T_b1 (ReceiveOrder)	接收订单	0.9
2	T_a1 (DealOrder)	上线接单	T_c4 (WaitOrder)	接单处理	0.65
3	T_a4 (ArriveAtMerchant)	骑手到达商家	T_b5 (RiderArrive)	骑手已到达	0.98
4	T_b6 (Complete_2)	交接餐品完成	T_c5 (GetFood)	接收到餐品	0.90

基于 DeepSeek 大语言模型的具体语义分析如下：

- 第 1 组中，Ta1 “上线接单” (骑手智能体)与 Tb1 “接收订单” (商家智能体)分别从骑手端与商家端描述同一业务起点，前者表示骑手进入工作状态准备接单，后者表示商家接收到用户订单，两者在业务流程中前后衔接、语义高度相关，因此相似度估算为 0.85~0.95，建议评分 0.90；
- 第 2 组中，Ta1 “上线接单” (骑手智能体)与 Tc4 “接单处理” (用户智能体)分别属于不同智能体的

不同流程阶段,前者为骑手 readiness 状态,后者为用户下单后系统进行接单处理的状态,两者分处流程上下游、语义存在一定关联但阶段差异明显,因此相似度估算为 0.60~0.70,建议评分 0.65;

- 第 3 组中, Ta4 “骑手到达商家”(骑手智能体)与 Tb5 “骑手已到达”(商家智能体)主体均为“骑手”,动作均为“到达”,对象均指向“商家”,语义高度一致,仅表述略有差异,且在业务流程中对应同一事件,因此相似度估算为 0.95~1.0,建议评分 0.98;
- 第 4 组中, Tb6 “交接餐品完成”(商家智能体)侧重商家完成交接动作,而 Tc5 “接收到餐品”(用户智能体)则从用户角度描述接收动作,两者为同一事件(交接与接收)的不同视角,语义关联较强,但由于“交接完成”隐含双方确认,“接收到”仅强调一方状态,语义并不完全等同,因此相似度估算为 0.85~0.95,建议评分 0.90。

基于上述识别出的共享变迁集(表 3),自动推导并执行“变迁合并、流关系重定向、初始标识统一”等合成规则,完成全局 Petri 网构建,合成后的 Petri 网如图 5 所示。

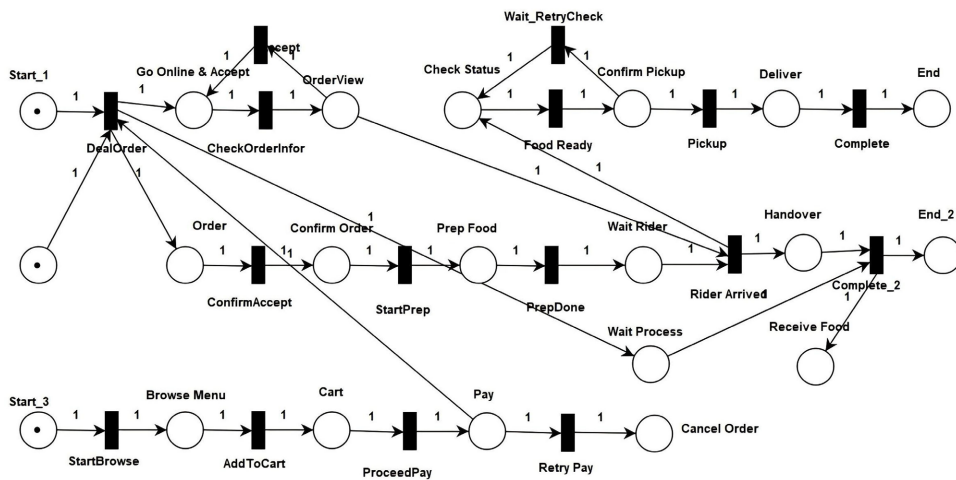


Figure 5. Petri net synthesized by LLM

图 5. LLM 合成后的 Petri 网

4.4. 模型验证及其结果

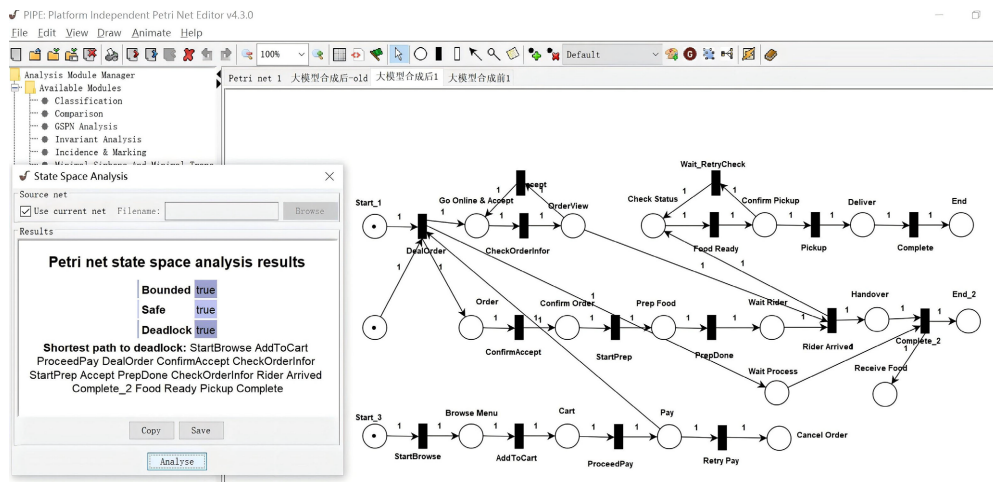


Figure 6. The model checking results by the PIPE tool

图 6. 基于 PIPE 工具模型检测结果

利用 PIPE 工具对合成后的全局 Petri 网进行模型检测, 检测结果如图 6 所示。检测表明, 该 Petri 网是有界且安全的, 系统在运行过程中不会出现状态爆炸及资源越界问题; 但检测同时发现系统中存在死锁, 说明当前服务交互设计存在缺陷, 可能导致业务流程无法正常终止或陷入僵局。

5. 结论

本文提出了一种融合大语言模型与 Petri 网的智能体服务组合方法。该方法首先采用 UML 对智能体服务流程进行统一建模, 进而通过 XMI 与 XSLT 技术实现 UML 模型向 Petri 网的自动化转换; 在此基础上, 创新性地引入大语言模型对 Petri 网进行语义标注, 并驱动基于语义匹配的共享合成, 最终利用 PIPE 工具对合成后的模型进行形式化验证。实验结果表明, 该方法通过大语言模型的语义理解能力取代了传统依赖人工的接口匹配规则, 有效提升了服务组合的智能化水平和自动化程度, 同时借助 Petri 网的严格数学基础与 PIPE 的模型检测能力, 保障了组合流程的行为正确性与可靠性, 为处理复杂、异构的智能体服务组合问题提供了一种行之有效的解决方案。

基金项目

浙江理工大学大学生创新训练计划项目(202410338054)。

参考文献

- [1] 李彬, 崔汉国, 李正民, 等. 基于共享子网合成 Petri 网的协同虚拟维修系统[J]. 中国修船, 2015, 28(5): 45-49.
- [2] 薛瑜, 李承家. 模糊 Petri 网的合成运算[J]. 杭州电子科技大学学报, 2012, 32(3): 79-82.
- [3] Desel, J. and Reisig, W. (1996) The Synthesis Problem of Petri Nets. *Acta Informatica*, **33**, 297-315. <https://doi.org/10.1007/s002360050046>
- [4] 刘之家, 李文敬, 王汝凉. Petri 网共享合成及其在并行系统中的应用[J]. 计算机工程与设计, 2011, 32(3): 968-971+983.
- [5] Mahadeer, A., Arulprakasam, R. and Dare, V.R. (2024) Partial Petri Net Languages and Their Properties. *International Journal of Information Technology*, **16**, 3663-3676. <https://doi.org/10.1007/s41870-024-01903-0>
- [6] Ronny, T., Susanna, D., Stefan, H., et al. (2022) The Complexity of Synthesis of B-Bounded Petri Nets. *Fundamenta Informaticae*, **183**, 125-167. <https://doi.org/10.3233/fi-2021-2084>
- [7] 欧毓毅, 王浩, 凌捷. 基于智能体的动态组合服务的研究[J]. 计算机应用与软件, 2008(8): 108-110+120.
- [8] 陈宇浩, 屠翰乾, 相东明. 基于解释 Petri 网的新型 CPS 分解方法[J]. 计算机科学, 2025, 52(4): 49-53.
- [9] 于绍琪, 田玉平. 基于 Petri 网与多智能体深度强化学习的 AGV 路径规划[J]. 控制与决策, 2025, 40(5): 1438-1446.
- [10] 赵杰民. 基于并行计算的 Petri 网可达状态研究[D]: [硕士学位论文]. 西安: 西安电子科技大学, 2024.